

PROCESS SQL

PROCESS SQL *ddm-name* <<*statement-string*>>

Function

The PROCESS SQL statement is used to issue SQL statements to the underlying database.

ddm-name

The name of a DDM must be specified to provide the "address" of the database to which the statement string (see below) is to be addressed. For more information, see ddm-name.

statement-string

The statements which can be specified in the *statement-string* are the same statements which can be issued with the SQL statement "EXECUTE" (see also Flexible SQL).

Note:

To avoid transaction synchronization problems between the Natural environment and the underlying database, the COMMIT and ROLLBACK statements must not be used within PROCESS SQL.

The statement string can cover several statement lines without any continuation character to be specified. Comments at the end of a line as well as entire comment lines are possible.

The statement string can also include parameters.

Parameters

$$\begin{bmatrix} :U \\ :G \end{bmatrix} : \textit{host-variable} \text{ [INDICATOR : } \textit{host-variable} \text{] [LINDICATOR : } \textit{host-variable} \text{]}$$

Unlike with the parameter described, in this context *host-variables* must be prefixed by a colon (:). In addition, they can be preceded by a further qualifier (":U" or ":G").

See further details on host-variable.

:U:host-variable

The prefix ":U" qualifies the host variable as a so-called "Using" variable. Such a variable indicates that its value is to be *passed to* the database. ":U" is the default specification.

:G:host-variable

The prefix ":G" qualifies the host variable as a so-called "Giving" variable. Such a variable indicates that it is to *receive* a value *from* the database.

Examples

Examples for DB2 (under OS/390):

```
PROCESS SQL DB2_DDM << CONNECT TO :LOCATION >>
```

```
PROCESS SQL DB2_DDM << SET :G:LOCATION = CURRENT SERVER >>
```

Example for Adabas D:

```
PROCESS SQL ADABAS_D_DDM << LOCK TABLE EMPLOYEES IN SHARE MODE >>
```

Example of Calling a Procedure Stored in Adabas D:

The called procedure computes the sum of two numbers.

```
...
COMPUTE #N1 = 1
COMPUTE #N2 = 2
COMPUTE #SUM = 0
...
PROCESS SQL ADABAS_D_DDM << DBPROCEDURE DEMO.SUM (:#N1, :#N2, :G:#SUM) >>
...
WRITE #N1 ' + ' #N2 ' = ' #SUM
...
```

ENTIRE ACCESS Options

With ENTIRE ACCESS, you can also specify the following as *statement-string*:

- **SET SQLOPTION** *option* = *value*
- **SQLCONNECT** *option* = *value*
- **SQLDISCONNECT**

These options are only possible with ENTIRE ACCESS, and are described in the ENTIRE ACCESS documentation.